TABES 94-3001

AUTOMATED TEST ENVIRONMENT FOR A REAL-TIME CONTROL SYSTEM

NAS8-4000

7N-63-CR

70480

by

Ronald O. Hall
Rockedyne Division
Rockwell International Corporation
Southeastern District Operations
Suite 1
555 Discovery Drive
Huntsville, Alabama 35806

ABSTRACT

An automated environment with hardware-in-the-loop has been developed by Rocketdyne Huntsville for test of a real-time control system. The target system of application is the man-rated real-time system which controls the Space Shuttle Main Engine(SSME). The primary use of the environment is software verification and validation, but it is also useful for evaluation and analysis of SSME avionics hardware and mathematical engine models. It provides a test bed for the integration of software and hardware.

The principles and skills upon which it operates may be applied to other target systems, such as those requiring hardware-in-the-loop simulation and control system development. Potential applications are in problem domains demanding highly reliable software systems requiring testing to formal requirements and verifying successful transition to/from off-nominal system states.

This paper will provide an overview of the environment and the advantages provided.

INTRODUCTION

The need for a high fidelity simulation facility was recognized early in the SSME program. Any undetected control system flaw could be disastrous, either on a test stand or shuttle orbiter flying a Space Transportation System(STS) mission. NASA/MSFC started the initial design of such a facility and the original version became operational at the MSFC in 1975. The facility provides a program unique capability for simulating SSME system operations at nominal and off-nominal conditions. The intent is to perturb variables of software, hardware, or internal engine operations to determine the SSME system response in all modes of operation.

Rocketdyne assumed responsibility for the facility, known as the Hardware Simulation Laboratory(HSL), in 1977. Rocketdyne's experience in operating the lab indicated the need for automation. As the system matured, an extensive test suite and testing process was developed to determine if the target system functions per requirements. The leverage automation could bring to the process became evident.

When the target system was enhanced to a Motorola 68000 series central processing unit, the go-ahead was given to build an automation environment into the HSL. This effort culminated in 1987 with the release of the first version of the environment. Subsequent versions have been released which have extended and refined the capability.

DISCUSSION

The laboratory contains a maximum complement of flight-type hardware (actuators, igniters, sensors, solenoids) and alternately uses component simulation. The SSME is represented with a high fidelity real-time software engine model. Computers induce in real-time off-nominal conditions through failure activation hardware. Test data is accumulated as testing progresses and is automatically analyzed later in non real-time.

ENVIRONMENT CONTROL SYSTEM

Test application is automatically controlled via a software system distributed across the laboratory. A unified collection of test tools allows the specification of tests and data reduction in a simple scripting language. Different versions of the target system may be tested in the environment with no change to the environment control software.

The heart of the system is a compiler and database. The compiler parses scripts generated by users and generates code for the various devices and for test results analysis. The database provides flexibility by maintaining definitions for devices and symbols required by the compiler. The resulting functionality is that of a state-of-the-art facility providing high density test histories generated in a flexible, repeatable, comprehensive test environment.

TEST ACTIVITY

The test activity consists of three distinct stages: test generation, test application and test results analysis. Test generation is an off-line process using a text editor to construct test cases in the scripting language. This process is essentially a mapping of the individual software requirements into test scripts. Failure scenarios are scripted and stored in files along with expected results.

Test application is the actual running of the tests in the laboratory. Test cases are grouped into logical divisions and run as units. Test results are captured for subsequent analysis.

Expected test results are embedded in the test scripts. Test results analysis is the off-line application of the expected results segment of the test scripts to the test results. Test results are automatically analyzed to determine if they are what is expected. Result logs are recorded in files to maintain documented evidence of whether the test results are as expected. The test activity flow is depicted in figure 1.

DEVELOPMENT SKILLS

The development of the environment has attracted and produced skills across the entire software life cycle, plus skills in hardware fabrication, design and maintenance, configuration management, MSFC security and quality control. Total quality management concepts are implemented, including the application of continuous process improvement activities and the application of process metrics. A wrap of engineering and programming expertise has been developed which is particularly applicable to the development and operation of automated test environments.

ADVANTAGES OF AUTOMATED TEST ENVIRONMENT

The primary benefit of the automated test environment is that it facilitates a better testing process. Testing is better in terms of test volume, throughput, flexibility, repeatability, exhaustive exercise of the target system and the precise control of the environment. Advantages range over the sub processes of test planning, test performance, test tracking and test management.

TEST PLANNING

The planning of the testing activity is facilitated by the way the environment structures the testing process. The work of test generation may be planned to support subsequent testing and analysis. Laboratory time may be scheduled as the queue of test cases ready for test increase. Test results analysis may be partitioned among personnel and workstations. Personnel and resources may be mixed and matched as the testing activity moves to completion.

TEST PERFORMANCE

The performance of testing is advantaged by the environment at all three stages of the testing activity. The flexibility and repeatability provided translate into a dramatic reduction of the test period. A 50% time savings for a complete end-to-end verification of the target system has been realized.

In the test generation stage the construction of test cases is made easy and natural via scripting: Software requirements are reformulated as test cases in the scripting language, in the vocabulary of the system. The compiler and database combine to make this possible. The database maintains meaningful names for device and software addresses.

These names are used in the test scripts in test case specification. The use of physical addresses is avoided, meaning the compiler is independent of laboratory hardware changes and shifting software addresses. Multiple environment definitions may be maintained in the database to allow the flexibility of using different laboratory configurations and versions of the target system. The functional level of the target system is addressed, avoiding the intricacies of the laboratory. The use of raw data are avoided, as data may be entered and expected results expressed in engineering units.

In the test application stage the environment reduces the scope of the work effort to be the application of selected test suites in selected time periods. In each session the environment is initialized to the appropriate laboratory configuration and target system version, allowing anomalous system states to be repeated as necessary. Thus questionable tests results may be quickly and easily duplicated. Technicians may run the suites, allowing test engineers to concentrate on test generation and analysis.

In the test results analysis stage data reduction is preprogrammed to determine if results are as expected. Engineer activity at this stage is usually reduced to scanning the output logs for the phrase "Results as expected." When results are not as expected failure tables are generated to show actual results.

TEST TRACKING

Tracking is essential to the test activity to provide visibility to management and customer. Tracking is advantaged by the environment in allowing a view of the test process in terms of what is being accomplished in relation to what is planned. The environment facilitates work progress to be quantified in terms of the essential outputs of the process: test cases/software requirements ready for test as well as those completely tested with results analyzed, and counts of test case lines of source code.

TEST MANAGEMENT

Test management is advantaged by the environment from the perspectives of target system validation and change management. The environment produces documented evidence of whether the target system functions per specification. This validation information is necessary to the formal release of the target system for field use.

As the target system evolves, change management becomes a central focus. Requirements are added, taken away and modified. The environment facilitates the configuration management of the changing test case suite over multiple releases of the target system. As the test cases reside in computer files, they may be managed in a code management system in the same manner as programming source code. Only those test cases need change whose corresponding software requirements change; reuse is facilitated. Additional test cases are added to the suite as new requirements are added.

CONCLUSIONS

This paper provides an overview of an automated test environment for a real-time control system. The advantages of such an environment are numerous. The principles and skills upon which it operates may be applied to other target systems such as those requiring hardware-in-the-loop simulation and control system development. Potential applications are in problem domains demanding highly reliable software systems requiring testing to formal requirements and verifying successful transition to/from off-nominal system states. Possible domains of interest are in such areas as aerospace, defense, nuclear power management and medical technology.

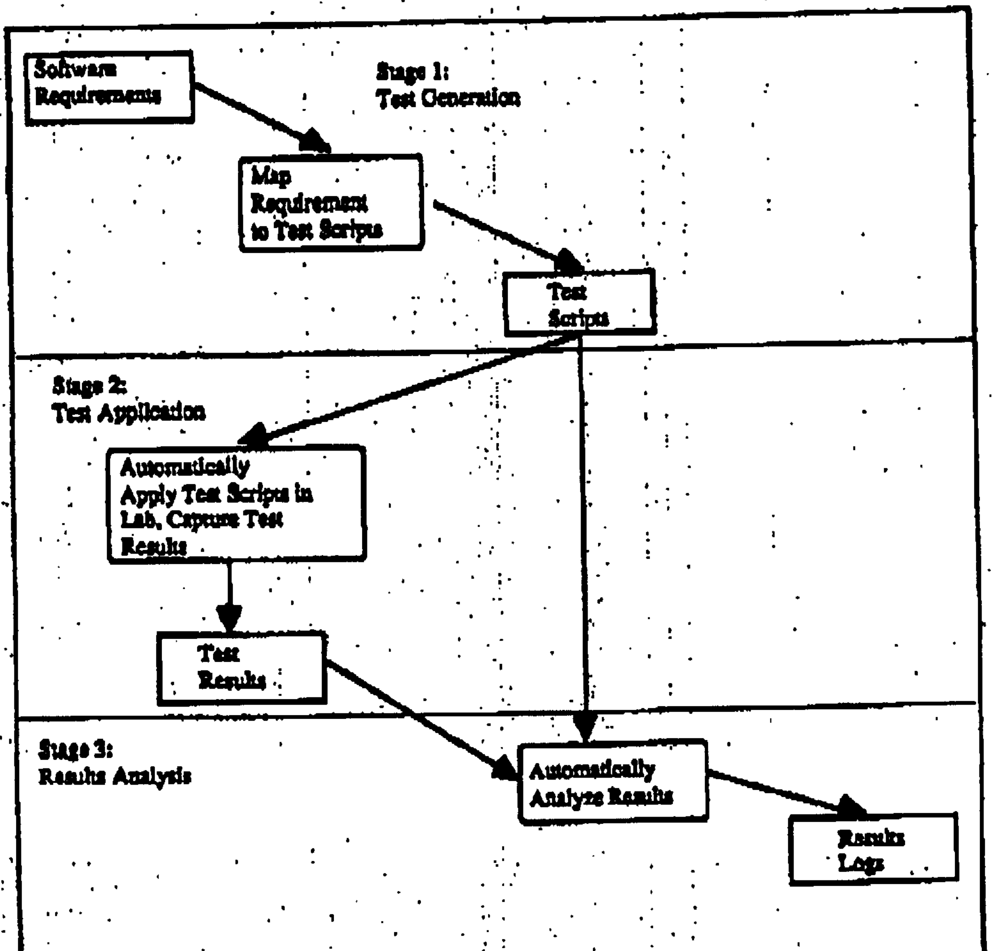

Figure 1. Test Activity Flow